

Iterated Singular Value Decomposition for Dimensionality Reduction

Sudip Khadka

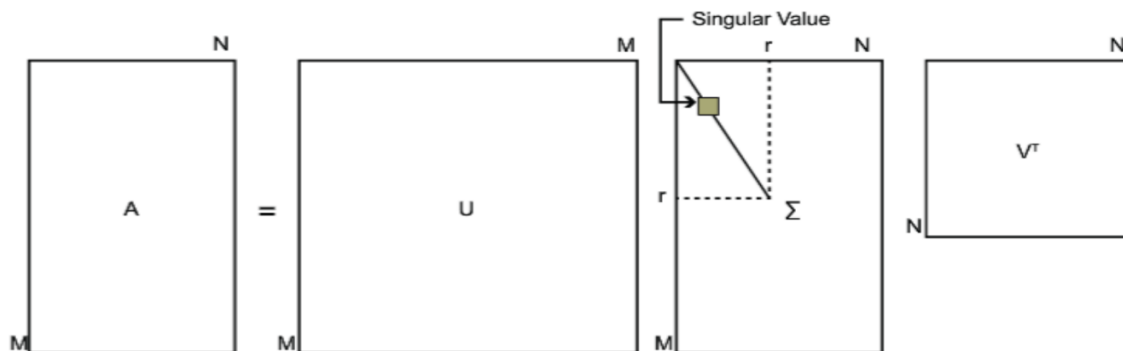
Any statistical and machine learning models are subject to the curse of dimensionality. One of the most sophisticated dimensionality reduction and anomaly detection techniques in the field of machine learning is the autoencoder. Despite its advantages, it has a major disadvantage; lack of transparency, often referred as the 'Black Box' problem. In contrast, I am employing one of the most powerful tools in matrix decomposition, which serves as a building block for machine learning, Singular Value Decomposition (SVD). I implemented an iterative method to reduce the dimensionality using the eigen decomposition and extracted the important features through the truncation and reduced the dimension as smaller as possible while maximizing the out of sample prediction via integrating logistic regression.

Singular Value Decomposition (SVD):

SVD is the matrix factorization technique which uses very simple and interpretable linear Algebra. It takes higher dimensional data and try to compress into lower dimensional while generalizing it in terms of its dominant patterns or correlation. SVD is used for a variety of purposes. In particular, we are interested in its application for dimensionality reduction to build models for tabular datasets. We are trying to uncover dominant combination of features that describe data as much as possible. Any matrix regardless of symmetry, shape and rank are unconditionally decomposed into three matrices.

$$\bar{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$$

Where,



\bar{A} is $m \times n$ matrix. It represents the higher dimensional dataset that needs to be transformed into lower dimensional. U is an $m \times m$ matrix that is orthogonal. It contains information of the column space of A i.e. importance each column. V is $n \times n$ matrix that is also orthogonal.

It contains the information of the row space of A . Σ is an $m \times n$ diagonal matrix with non-negative real number that tells how important the columns of U and V are. Each singular values are arranged in hierarchical order i.e. $\sigma_1 > \sigma_2 > \dots > \sigma_n > 0$

Proposed Idea: Reducing Dimension through iteration.

Algorithm:

For iteration in range(R):

iteration=1

Step1: Decompose \bar{A} into $U \Sigma V^T$

Step2: Project into lower dimensions (3D) (analyze the clustering)

Step3: Take N singular vectors that explain at least 90% of variance.

Step4: Use Truncation¹ to get the feature scores.

Step5: Take first N important features and evaluate the performance using logistic Regression (optimize)

iteration =2

Step1: Create new matrix $\bar{A}1$ with features vectors that maximized the accuracy in first Iteration, where $\bar{A}1 \subseteq \bar{A}$

Step2: Decompose $\bar{A}1$ into $U \Sigma V^T$

Step3: Project into lower dimensions (3D) (analyze the clustering)

Step4: Take first N singular vectors that explain at least 90% of variance.

Step5: Use Truncation to get the feature scores.

Step6: Take N important features and evaluate the performance using logistic Regression (optimize)

iteration = n

Repeat with new matrix.

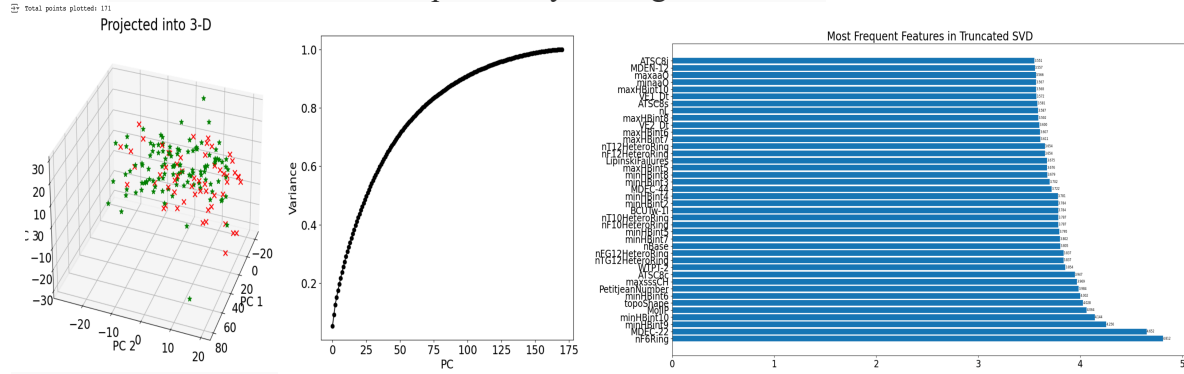
Number of Iteration depends on the dimension and the complexity of the data. For instance, in this example I have the dataset that includes 171 molecules designed for functional domains of a core clock protein, CRY1, responsible for generating circadian rhythm. 56 of the molecules are toxic and the rest are non-toxic. The shape of the data frame is (171, 1203) i.e. it has 171 observations and 1203 features vectors. The goal is to reduce dimensionality while improving model accuracy. This approach predicts whether the molecules are toxic or non-toxic using logistic regression.

¹ Truncated SVD assumes the features that have the highest coefficients in the singular vector considered the most important features.

Results:

First Iteration $\bar{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$:

= 90% of the variance is explained by 97 singular vectors.



The performance of the model is evaluated based on precision², recall³ and f1-score. With first 100 important features from 97 singular vectors we get the model prediction accuracy of 73% and f1-score⁴ of 71% with the following classification matrices:

	precision	recall	f1-score	support
0	0.91	0.67	0.77	15
1	0.29	0.67	0.40	3
accuracy			0.67	18
macro avg	0.60	0.67	0.58	18
weighted avg	0.81	0.67	0.71	18

Second Iteration $\bar{A}_1 = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$:

= 90% of the variance is explained by 58 singular vectors.

² Precision measures how well the model guess the label of interest; let's say True Positive. It is given by

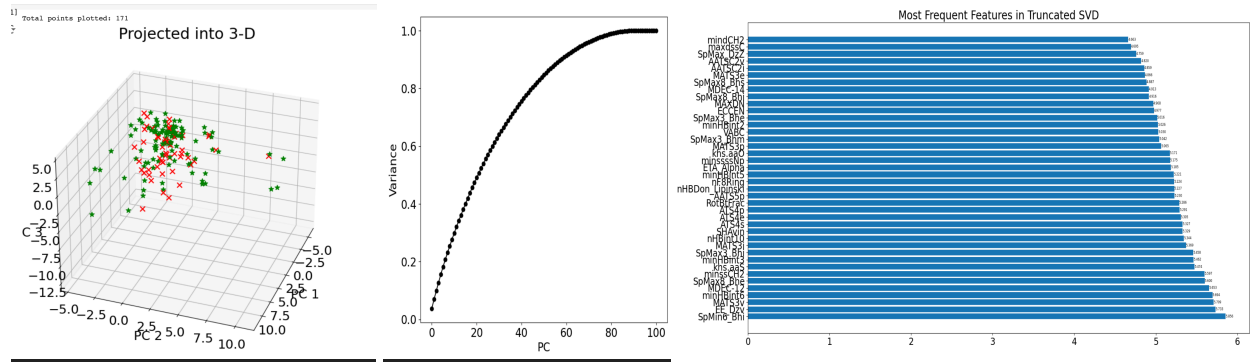
$$= \frac{\text{Correct Positive Guess}}{\text{Total Positive Guess}} = \frac{TP}{TP + FP}$$

³ Unlike Precision it takes negative labels into equation to optimization.

$$= \frac{\text{Correct Positive Guess}}{\text{All Positive Guess}} = \frac{TP}{TP + FN}$$

⁴F1-score evaluate how good is the quality of prediction and how completely have the model predicted the label.

$$= \frac{2 * \text{Precision} * \text{recall}}{\text{Precision} + \text{recall}} = \frac{2 * TP}{2 * TP + FP + FN}$$

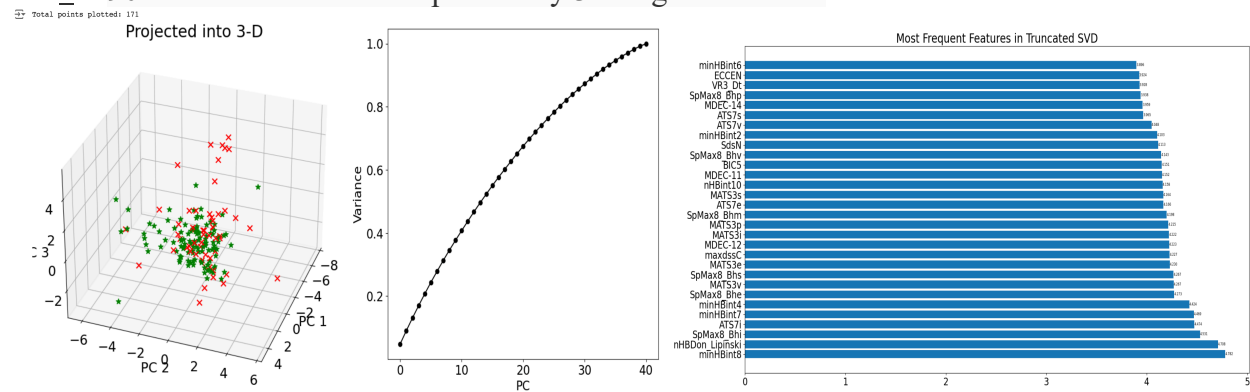


With first 40 importants features from 58 singular vectors we get the model prediction accuracy of 88.8% with the following calssification matrices:


	precision	recall	f1-score	support
0	1.00	0.83	0.91	12
1	0.75	1.00	0.86	6
accuracy			0.89	18
macro avg	0.88	0.92	0.88	18
weighted avg	0.92	0.89	0.89	18

Third Iteration $\bar{A}_2 = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$:

= 90% of the variance is explained by 32 singular vectors.



With first 15 importants features from 32 singular vectors we get the model prediction accuracy of 83.3% with the following calssification matrices :

		precision	recall	f1-score	support
	0	0.80	1.00	0.89	12
	1	1.00	0.50	0.67	6
	accuracy			0.83	18
	macro avg	0.90	0.75	0.78	18
	weighted avg	0.87	0.83	0.81	18

Conclusion:

Dimension reduction can be easily achieved implementing PCA through the Python sklearn package. However, first calculating the SVD provides more flexibility in visualizing projections in lower dimension and allows for truncation to assess feature importance. In our example, we reduced the dimension from (171, 1203) to (171,15) while achieving a out-of-sample prediction accuracy of 83% and a weighted average f1-score of 81%. In a second iteration, we achieved an accuracy of 88.8% and a weighted average f1-score of 89% with 40 features. Given that the dataset is specifically designed for a dimensional reduction challenge, an accuracy of 83% and a weighted average f1-score of 81% with 15 feature vector is considered good.

References

- Dorani, E. (2023, June 8). *How to plot feature importance using truncated SVD*. DEV Community. <https://dev.to/elldora/how-to-plot-feature-importance-using-truncated-svd-166n#:~:text=In%20truncated%20SVD%2C%20the%20singular,important%20features%20in%20the%20dataset.>
- Smilowitz, D. (n.d.). *Singular Value Decomposition*. RPubs.<https://rpubs.com/dsmilo/DATA643-Project3>
- Brunton, S. L., & Kutz, J. N. (2022). *Data-driven science and engineering machine learning, Dynamical Systems, and Control*. Cambridge University Press.
- Gül, Şeref and RAHİM,FATİH. (2022). Toxicity. UCI Machine Learning Repository. <https://doi.org/10.24432/C59313>.